WHAT IS CLAIMED IS:

1. For use in a processor, an instruction handling system for determining instruction folding comprising:

at least one fold decoder associated with an instruction fetch buffer stack,

the at least one fold decoder coupled to a set of successive entries within the instruction fetch buffer stack and examining contents within the successive entries prior to a main decode of the contents within the successive entries to determine whether the successive entries contain two or more instructions which may be folded,

the at least one fold decoder generating fold-status information for the contents within the successive entries indicating whether the successive entries contain two or more instructions which may be folded.

2. The instruction handling system as set forth in Claim 1 wherein the at least one fold decoder further comprises:

a plurality of fold decoders associated with the instruction fetch buffer stack and including the at least one fold decoder,

each fold decoder coupled to a different set of successive entries within the instruction fetch buffer stack, wherein the different sets of successive entries overlap, and examining contents within a corresponding set of successive entries to determine whether the corresponding set of successive entries contain two or more instructions which may be folded,

each fold decoder generating fold-status information for the contents within the corresponding set of successive entries indicating whether the corresponding set of successive entries contain two or more instructions which may be folded.

3. The instruction handling system as set forth in Claim 2 wherein the fold-status information produced by each fold decoder includes a number of instructions which may be folded and a size of each instruction which may be folded.

4. The instruction handling system as set forth in Claim 2 wherein the fold-status information for each set of successive entries is stored in association with the respective set of successive entries within the instruction fetch buffer stack.

5. The instruction handling system as set forth in Claim 1 wherein the at least one fold decoder checks the contents within the successive entries for instructions of a variable size and for possible folding of a variable number of instructions.

6. The instruction handling system as set forth in Claim 1 further comprising:

a decoder receiving the fold-status information together with the contents of the successive entries for translation of the contents of the successive entries into signals which may be operated on by an execution unit.

7. The instruction handling system as set forth in Claim 1 wherein the decoder employs the fold-status information during folding of at least the contents of the successive entries into a single operation.

8. A processor comprising:

an instruction fetch mechanism retrieving instructions for storage within an instruction fetch buffer;

an instruction decode mechanism for translating instructions into signals which may be operated on by at least one execution unit; and

an instruction handling system coupled between the instruction fetch buffer and instruction decode mechanism for determining instruction folding comprising:

at least one fold decoder associated with an instruction fetch buffer stack,

the at least one fold decoder coupled to a set of successive entries within the instruction fetch buffer stack and examining contents within the successive entries prior to a main decode of the contents within the successive entries to determine whether the successive entries contain two or more instructions which may be folded,

the at least one fold decoder generating fold-status information for the contents within the successive entries indicating whether the successive entries contain two or more instructions which may be folded.

9. The processor as set forth in Claim 8 wherein the at least one fold decoder further comprises:

a plurality of fold decoders associated with the instruction fetch buffer stack and including the at least one fold decoder,

each fold decoder coupled to a different set of successive entries within the instruction fetch buffer stack, wherein the different sets of successive entries overlap, and examining contents within a corresponding set of successive entries to determine whether the corresponding set of successive entries contain two or more instructions which may be folded,

each fold decoder generating fold-status information for the contents within the corresponding set of successive entries indicating whether the corresponding set of successive entries contain two or more instructions which may be folded.

10. The processor as set forth in Claim 9 wherein the fold-status information produced by each fold decoder includes a number of instructions which may be folded and a size of each instruction which may be folded.

11. The processor as set forth in Claim 9 wherein the fold-status information for each set of successive entries is stored in association with the respective set of successive entries within the instruction fetch buffer stack.

12. The processor as set forth in Claim 8 wherein the at least one fold decoder checks the contents within the successive entries for instructions of a variable size and for possible folding of a variable number of instructions.

13. The processor as set forth in Claim 8 wherein the instruction decode mechanism receives the fold-status information together with the contents of the successive entries.

14. The processor as set forth in Claim 8 wherein the instruction decode mechanism employs the fold-status information during folding of at least the contents of the successive entries into a single operation.

15. For use in a processor, a method of determining instruction folding comprising:

prior to decoding contents within a set of successive entries within an instruction fetch buffer stack,

examining the contents within the successive entries to determine whether the successive entries contain two or more instructions which may be folded; and

generating fold-status information for the contents within the successive entries indicating whether the successive entries contain two or more instructions which may be folded.

16. The method as set forth in Claim 15 wherein the step of examining the contents within the successive entries to determine whether the successive entries contain two or more instructions which may be folded further comprises:

examining contents within each of a different set of successive entries within the instruction fetch buffer stack, wherein the different sets of successive entries overlap, to determine whether the corresponding set of successive entries contain two or more instructions which may be folded.

17. The method as set forth in Claim 16 wherein the step of generating fold-status information for the contents within the successive entries indicating whether the successive entries contain two or more instructions which may be folded further comprises:

generating fold-status information for the contents within each set of successive entries indicating whether the corresponding set of successive entries contain two or more instructions which may be folded, wherein the fold-status information includes a number of instructions which may be folded and a size of each instruction which may be folded.

18. The method as set forth in Claim 16 further comprising:

storing the fold-status information for each set of successive entries in association with the respective set of successive entries within the instruction fetch buffer stack.

19. The method as set forth in Claim 15 wherein the step of examining contents within each of a different set of successive entries within the instruction fetch buffer stack further comprises:

checking the contents within the successive entries for instructions of a variable size and for possible folding of a variable number of instructions.

20. The method as set forth in Claim 15 further comprising:

transmitting the fold-status information together with the contents of the successive entries to an instruction decoder translating the contents of the successive entries into signals which may be operated on by an execution unit; and

employing the fold-status information during folding of at least the contents of the successive entries into a single operation within the instruction decoder.